

RAVEN AS PROBABILISTIC RISK ASSESMENT TOOL FOR RELAP5-3D

C. Rabiti

D. Mandelli, C. Smith, J. Schroeder, T. Riley, S. St
Germain, A. Alfonsi, J. Nielsen, J. Cogliati, R.
Kinoshita

www.inl.gov



IRUG September 13th 2013

RAVEN

REACTOR ANALYSIS AND VIRTUAL CONTROL ENVIRONMENT

The Supporting Program

The **Light Water Reactor Sustainability** Program (LWRS) defines a new methodological approach to meet the sustainability of the aging reactor fleet

RISMC

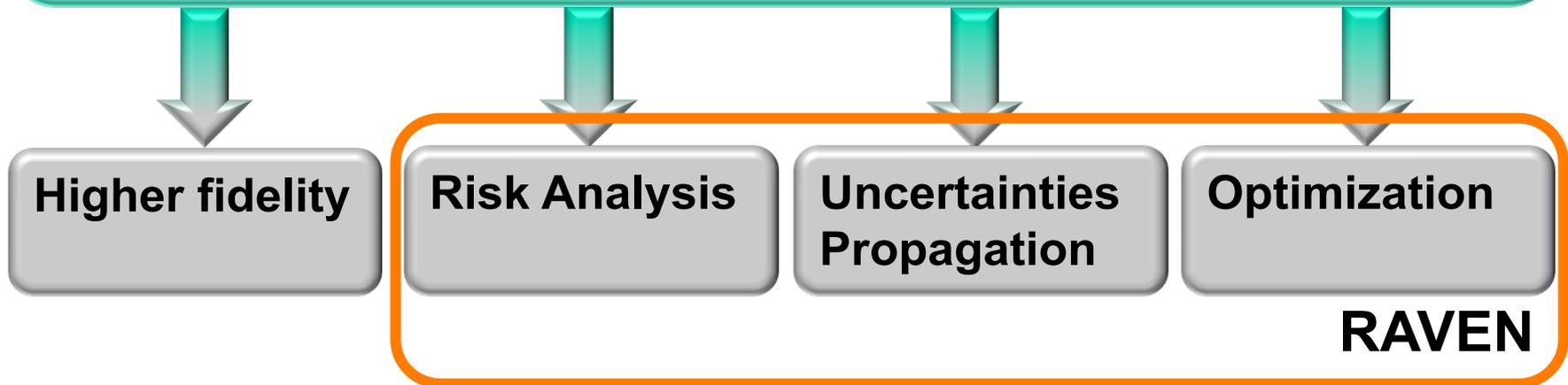
Risk Informed Safety Margins Characterization

The applications of RISMC needs capabilities that are not available in current tools

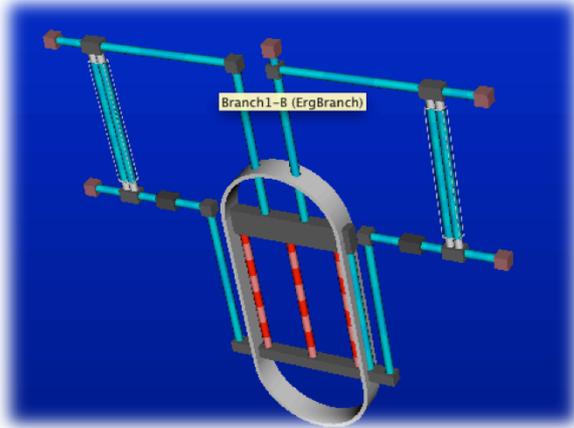
A new set of codes is therefore under development:
RAVEN, RELAP-7, GRIZZLY

Risk Informed Safety Margins Characterization (RISMC)

RISMC is a methodology to support plant decisions for risk-informed margin management with the aim to improve economics, reliability, and sustain safety of current NPPs
-NEEDS-

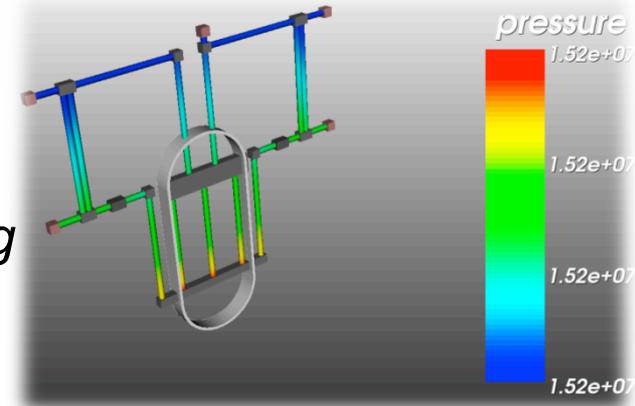


It started as a GUI for RELAP-7

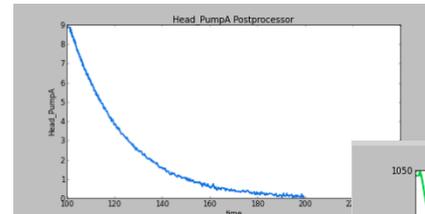
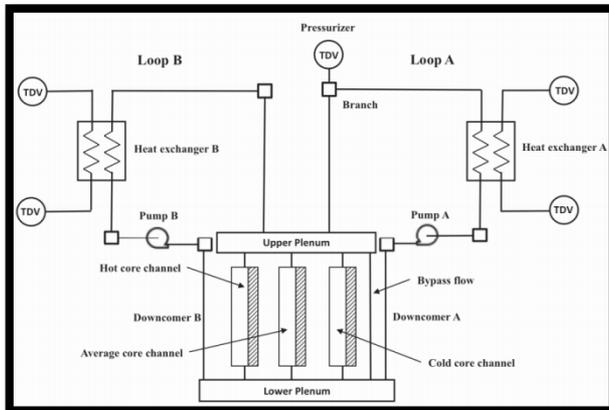


Interactive 3D view
of the plant

Online monitoring
of the solution

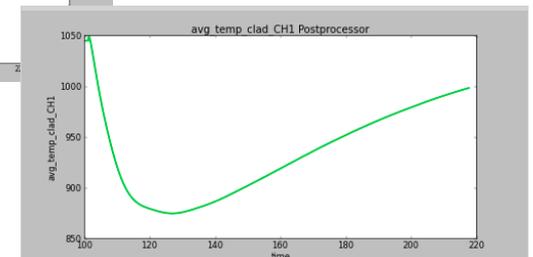


An increase in the
modeling capabilities should be paired with
more powerful data management tools to help
deployment



Pump head

Online monitoring
of plant signals



Clad
temperature

Risk Analysis

Uncertainty in system response

Uncertainty in initial condition

Probability distribution of the final status

Risk Analysis

**Risk =
Probability x Consequences**

**-Large Span of Scenarios
-Low Probability high consequences (?)**

RAVEN

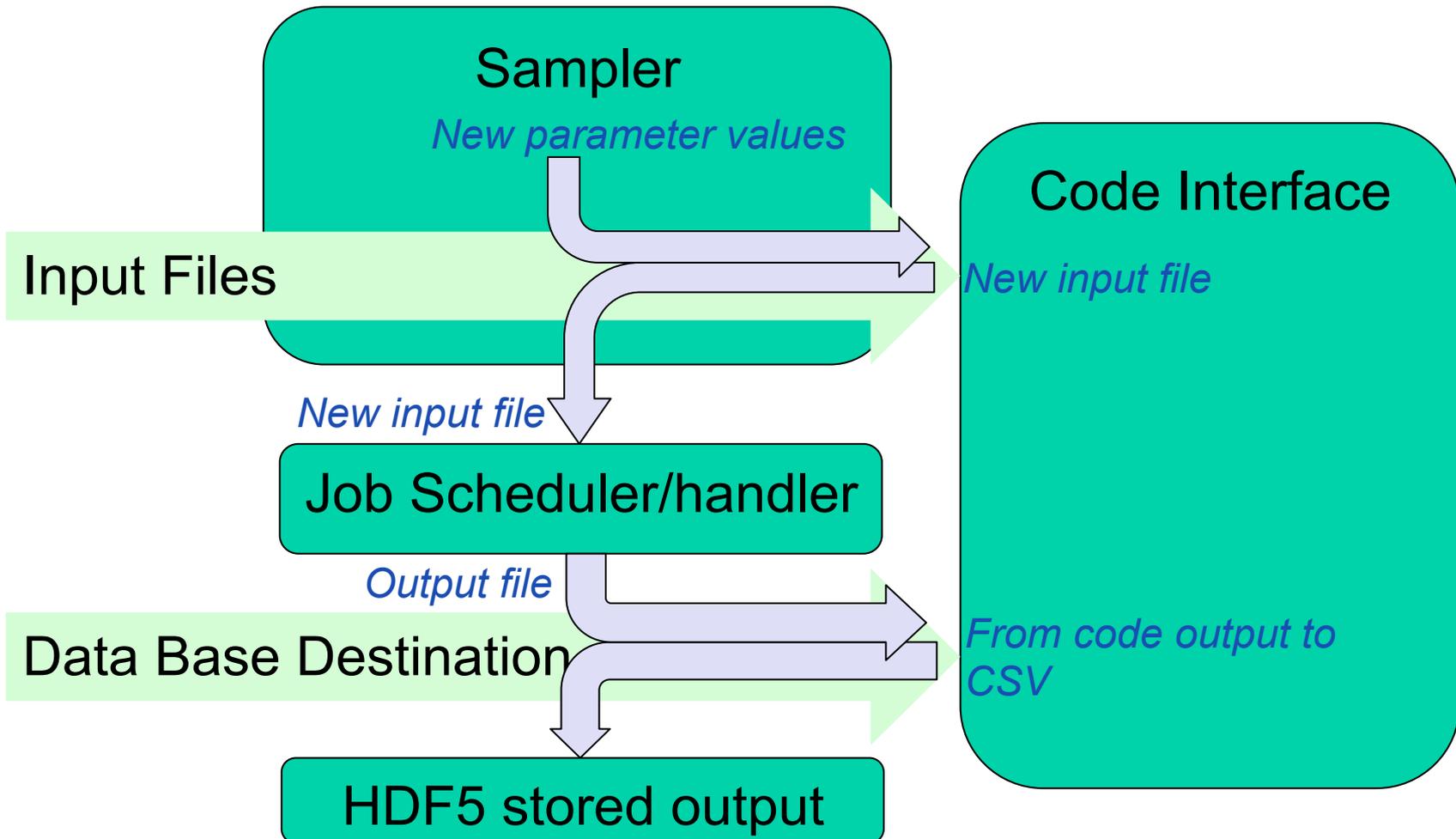
**Parallel
Monte-Carlo**

**Dynamic
Event Trees**

**Adaptive
Sampling**

The RAVEN Framework

- RAVEN implement a framework that is code agnostic...



Few Points that Might Have Been Unnoticed

- Any code could be added to the framework as long as
 - An interface to inject changes in the input file is provided
 - A csv output file is provided (conversion from CSV to HDF5 is provided internally)
- There are already few available type of sampling:
 - Monte Carlo
 - Latin Hyper Cube
 - Dynamic Event Tree (only RELAP-7 working in progress for RELAP5-3D)
- There are several type of distribution that could be used:
 - Normal (also truncated)
 - Triangular
 - Flat
 - More... (al distributions in scipy.stat)

Sampling and distribution could be used for any code only the interface code interface needs to be rewritten

...And Some More

- The Job scheduler runs in parallel (tested up to 10,000 runs in batches of hundreds)
 - The total max number of simultaneous runs is the batch size
 - The batch is continuously refilled up to the end of the requested simulations
 - Works with interactive session and PBS professional queuing software
 - The PBS interface is an class that could be overwritten to allow different queuing software
- Any type of post processing, graphical analysis will be based on the internal storing of data so that could be used for any code...

Not everything is already in place but it is going to be once for all (codes)

Next Step

All tasks (risk analysis, uncertainty quantification, and optimization) share a common challenges:

large number of simulations of the original problem are necessary to achieve the needed evaluations

- PRA needs a large number of simulation to know the probability distribution especially in low probability, high risk regions
- Uncertainty quantification requires the propagation of the probability distribution of the physical parameters
- Optimization requires the search for global minimum/maximum that are very expensive in terms of number of simulation required

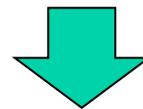
Luckily common problem have common approaches (might be not solution thus...)

Surrogate Models

- Not for all possible input but rather for a range of possibilities...
- Not for all possible geometries but for a parametric range...
- Not to represent all output space but rather some of the output...



It is possible to construct a surrogate model that reproduces a reasonable approximation of the reality that...



- Could be evaluated faster than the original set
- The parameters of the surrogate model set of equations could be *trained* to approximate the original system
- Most of the time the number of equations of the surrogate model can be arbitrary increased to achieve the sought accuracy

Really?

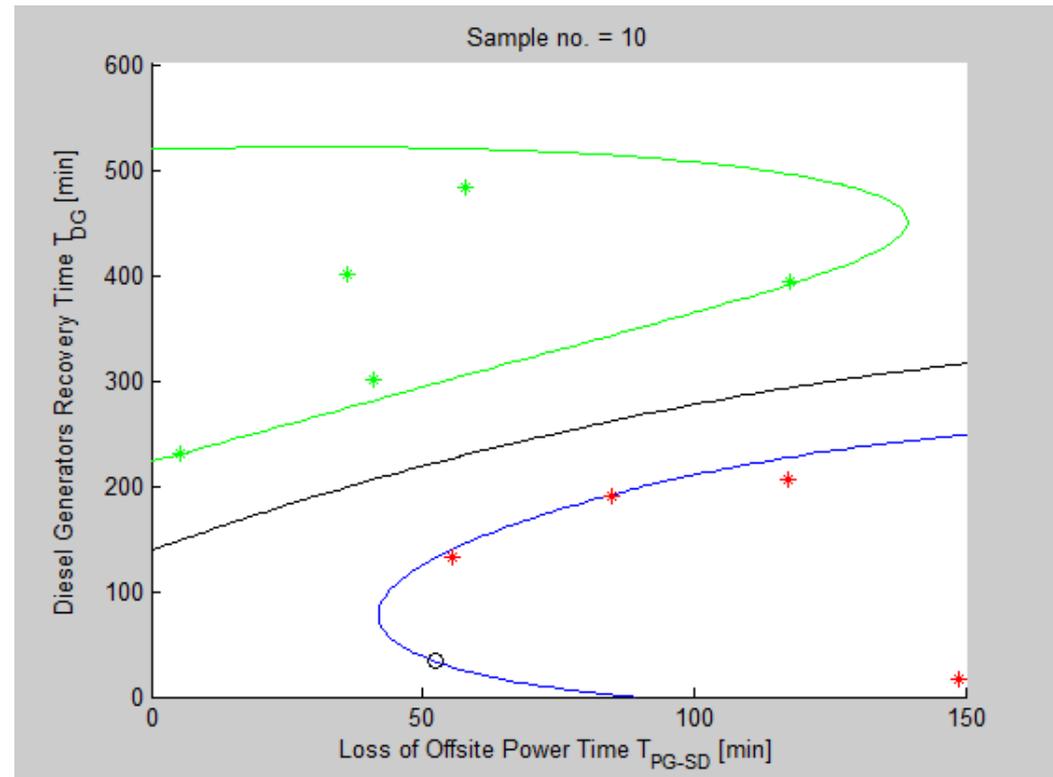
Why it is possible to construct a surrogate?

- Are all six equations needed when modeling single-phase fluid?
- If we want to know the max temperature do we need to know the temperature everywhere?
- Do we care if it is raining in Moscow and we want to go camping in Yellowstone?

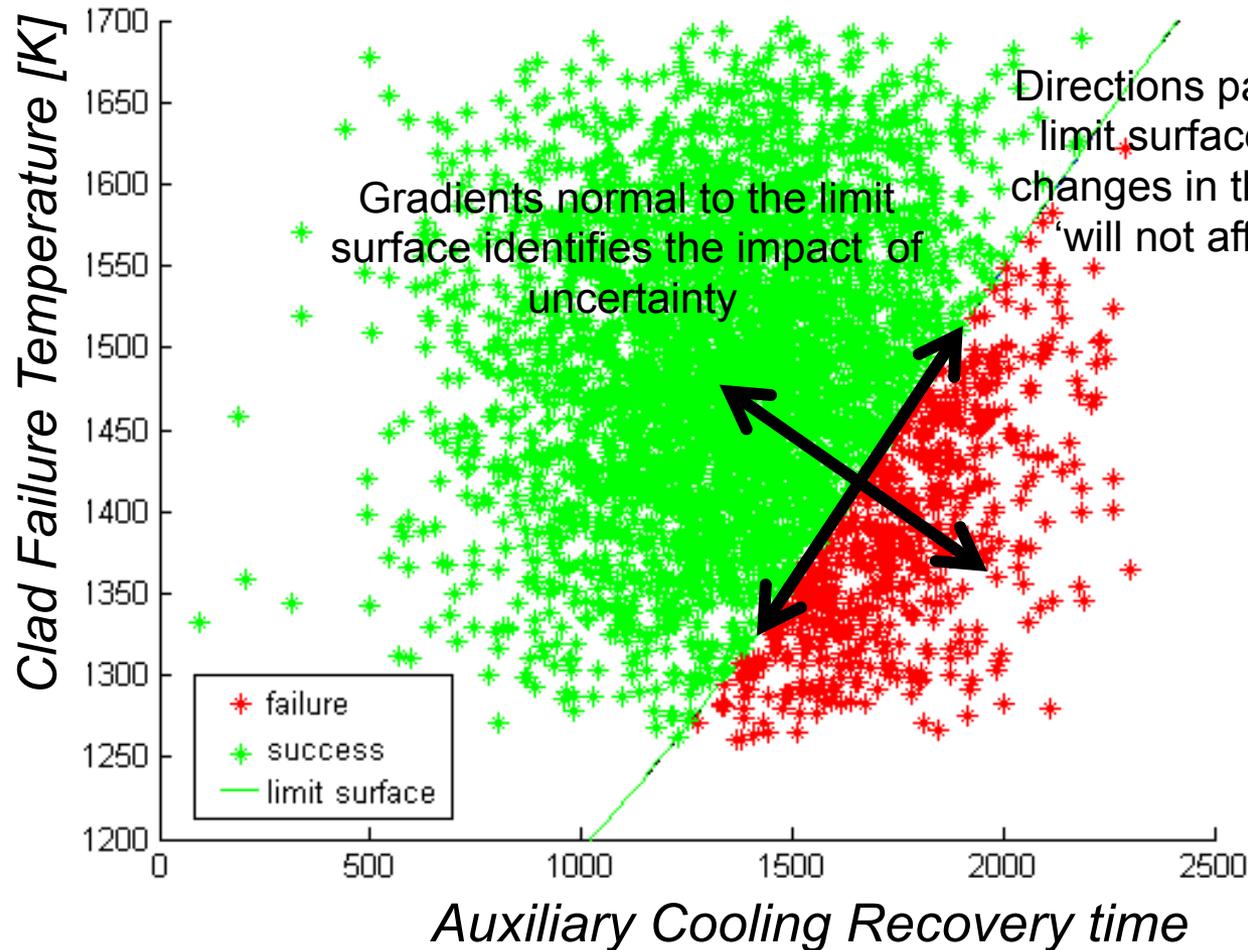
Our specific model lead to a subset of the possible solutions, this allows us the shrink the solution space and the degree of freedom of the mathematical representation of the problem

A Surrogate Model for the Limit Surface

- Classical surrogate models for the construction of limit surfaces are Support Vector Machines (SVMs)
- Our reduced space output is now the limit surface that we want to determine
- The goal of the SVM is to approximate the model in the surrounding of the limit surface
- Even better, the SVM decide if it is accurate enough, and if needs to be improved run the code where information is missing



The Limit Surface Example as the Output Space

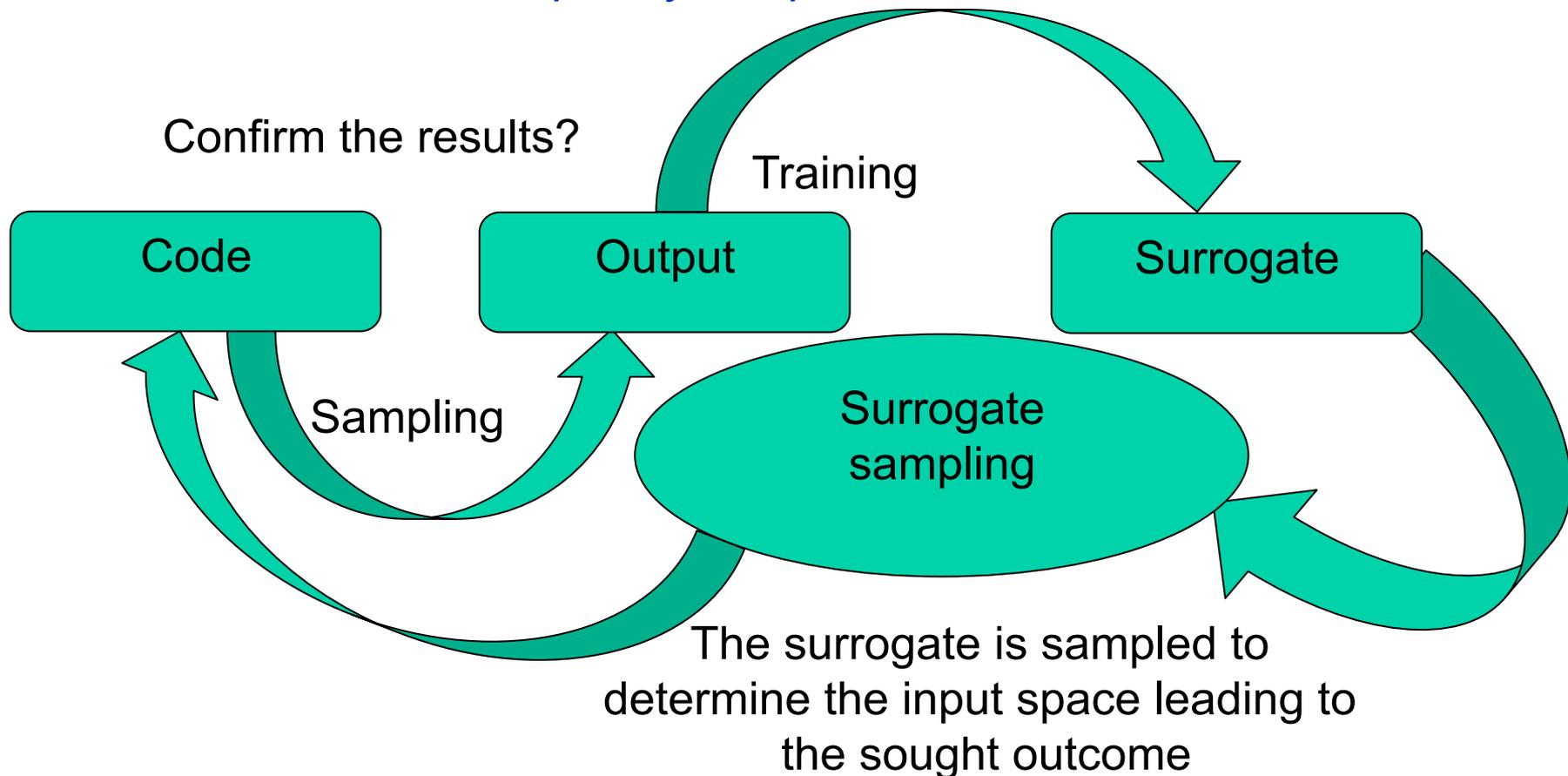


Limit surfaces are mathematical constructs that separate the input space between regions leading to success or failure

How RAVEN Generalize the Concept

The implementation is completely general (currently ongoing)

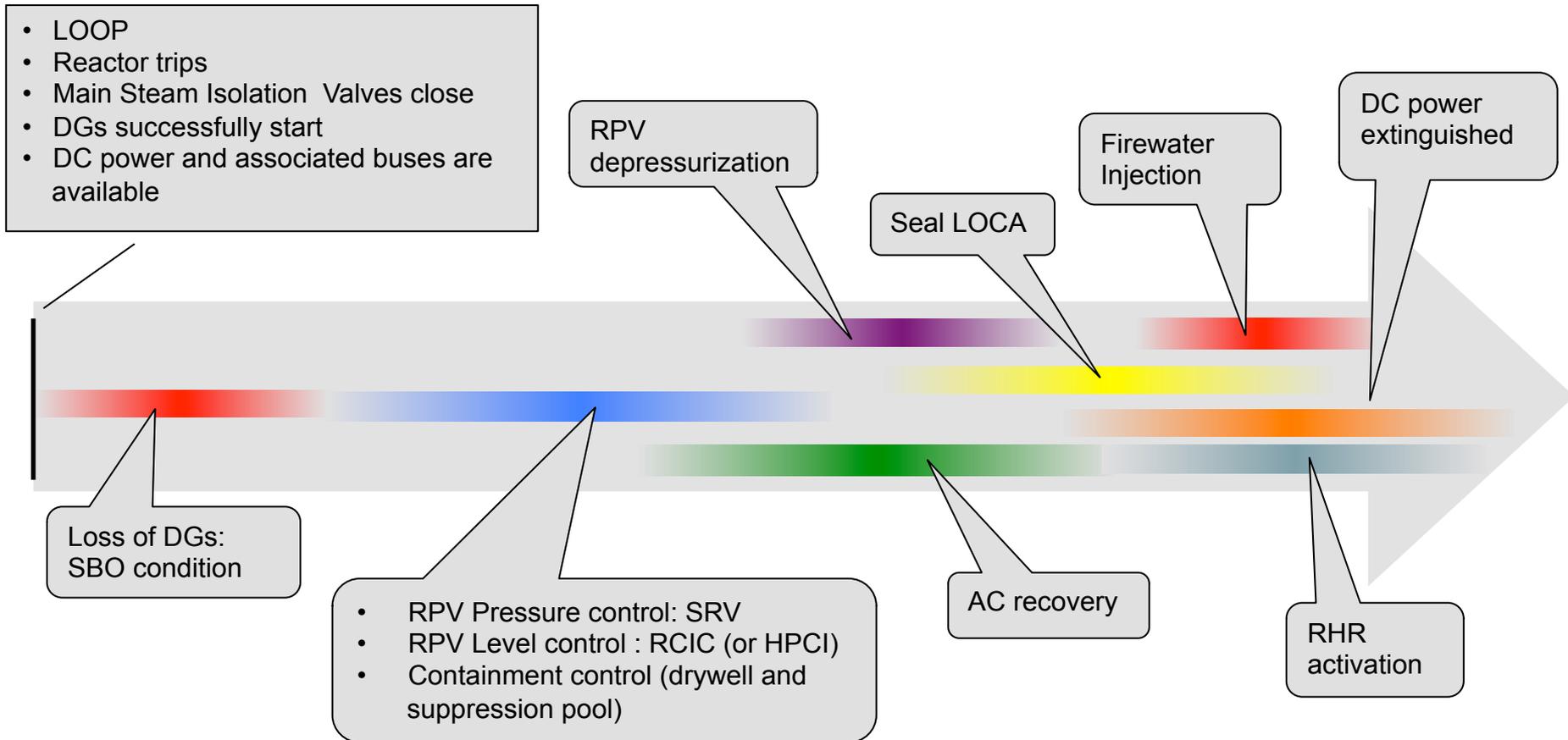
Once more, completely independent of the code used



Where we are...

- We are currently testing different software layout so to include post processing capabilities and very flexible convergence checks
- Of course the surrogate models available in literature are several hundreds, we are not re-writing all of them
 - Scikit.learn has been already interfaced with the framework (more than hundreds of surrogate models)
 - Working on a cooperation with Sandia to bring in DAKOTA
- Just a note (just to know how this methods are called in reality):
 - Supervised learning → a surrogate model relating input to output
 - Unsupervised learning → a surrogate model that binds region of the outputs by some relationship

Test Case: Scenario Example



Simulation Info

- Monte-Carlo analysis:
 - About 20,000 SBO sequences
- Stopping conditions:
 - AC power restored
 - Max clad temperature reached
- No severe accident analysis considered
- Ranking determined using SAPHIRE:
 1. Cut sets importance
 - Fussell-Vesely
 - Birnbaum
 - Uncertainty
 2. Event Tree structure

#	Parameter
1	Failure time of DGs
2	Recovery time of DGs
3	Battery life
4	SRV(s) fails open
5	Offsite AC power recovery
6	Clad Fail temperature
7	Containment fail pressure
8	Seal LOCA time
9	Seal LOCA flow rate
10	HPCI fails to run
11	RCIC fails to run

RAVEN Input Example

```
<Models>
  <Code name='MyRELAP' type='RELAP5'>python RELAP5run.py</Code>
```

We specify that we are using RELAP5-3D

```
</Models>
```

```
<Distributions>
```

```
<Uniform name='ECCSbattery'>
```

```
<hi>14400.0</hi>
```

```
<low>0.0</low>
```

```
</Uniform>
```

```
<Triangular name='Cladfailure'>
```

```
<apex>1477.0</apex>
```

```
<min>1255.0</min>
```

```
<max>1700.0</max>
```

```
</Triangular>
```

Here the list of distribution we would like to use

```
</Distributions>
```

```
<Samplers>
```

Here how a MC sampler will use it

```
<MonteCarlo name='TestMonteCarlo' limit='128'>
```

```
<Variable type='Uniform' distName='ECCSbattery' position='6'>0000588</Variable>
```

```
<Variable type='Uniform' distName='EDGrecovery' position='1'>0000204</Variable>
```

```
<Variable type='Uniform' distName='Cladfailure' position='6'>0000503</Variable>
```

```
<Variable type='Uniform' distName='PumpSealLOCATime' position='6'>0000599</Variable>
```

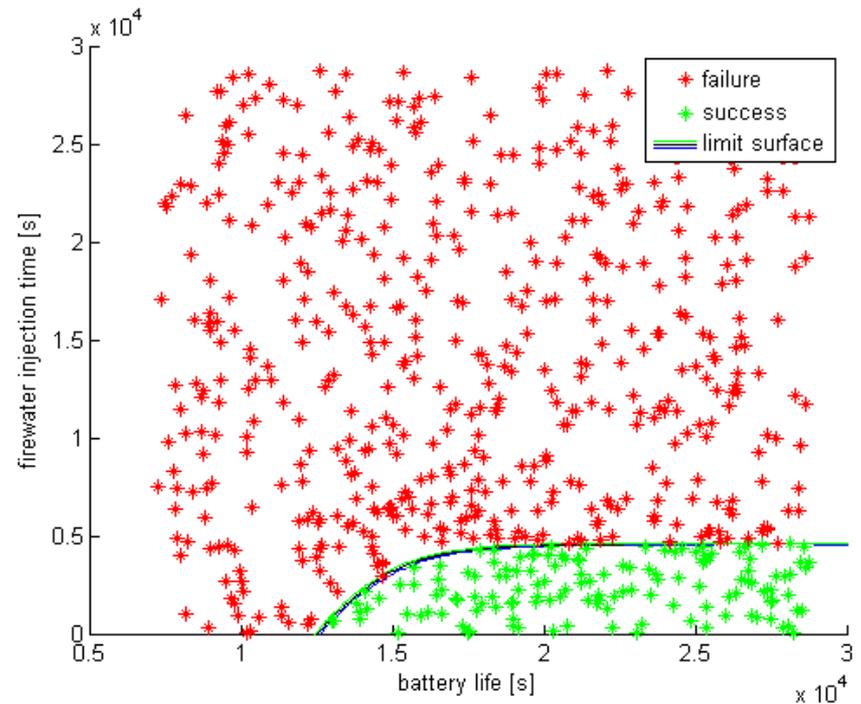
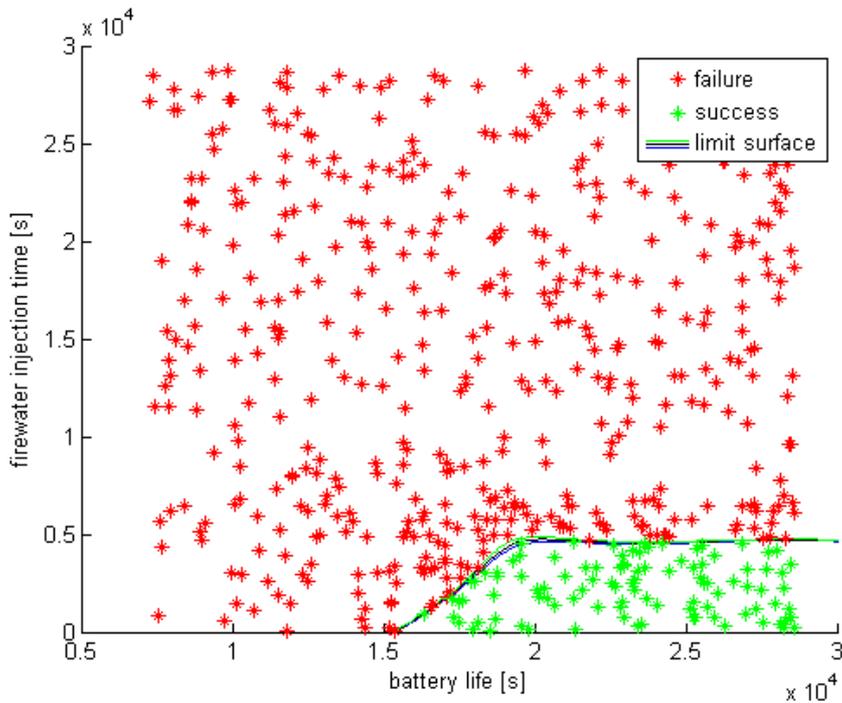
```
<Variable type='Uniform' distName='PumpSealLOCAArea' position='3'>7520101</Variable>
```

```
</MonteCarlo>
```

```
</Samplers>
```

Status

- 10,000 runs completed
- Post processing ongoing
- Very low number of RELAP-5 spurious failure



Conclusion

- RAVEN is a new tool to implement the RISMC concept providing:
 - UQ
 - PRA
 - Optimization
- The flexibility of the framework build makes it usable for RELAP-7 but also RELAP5-3D
- GUI, and a full implementation of the software infrastructure for surrogate models in the short list and probably ready before the end of 2013